1 545 169

PATENT SPECIFICATION

(11) 1 545 169

(21) Application No. 39465/77 (22) Filed 22 Sept. 1977
(44) Complete Specification published 2 May 1979
(51) INT CL² G06F 11/00
(52) Index at acceptance
G4A 12D 12N 16J 17B 17P 1C 5B 9C 9D EP

(72) Inventors ROBERT THOMAS MONTFORD GOULD
JAMES CHARLES McANLIS and
KULDIP KUMAR

(19)

(54) DATA PROCESSOR SYSTEM INCLUDING DATA-SAVE CONTROLLER FOR PROTECTION AGAINST LOSS OF VOLATILE MEMORY INFORMATION DURING POWER FAILURE

(71) We, BURROUGHS CORPORATION, a corporation of the State of Michigan, United States of America, of Burroughs Place, Detroit, Michigan 48232, United States of America, do hereby declare this invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:—

The present invention relates to data processor systems, and particularly to such systems protected against loss of volatile memory information in the event of a failure in the main power supply.

Various techniques have been devised in data processor systems to provide protection against loss of volatile memory information during a power failure. Some known techniques include an arrangement which, upon the detection of a power failure, permit the completion of the memory cycle which has already been initiated, but disable further memory cycles. In other known techniques, standby power is provided for an interval following the failure of the main power supply to protect the stored information and to permit continued operation. In still other systems, the memory contents are "dumped" into a dump memory, such as a tape cartridge, to thereby preserve the information until the power is restored. In a further known system, the data processor is programmed to provide an automatic interrupt upon the occurrence of any one of a number of specified conditions, one being the loss of primary power to the computer, during which conditions the control of the processor is transferred from a Normal Mode to a Control Mode which calls for the particular subroutine to be followed for the particular interrupt condition.

An object of the present invention is to provide a data processing system with a data-save controller which may be used for protecting the contents of the volatile registers of the processor against loss of information in the event of a failure in the main power supply.

According to a broad aspect of the present invention, there is provided a data processing system, comprising: a processor supplied with power from a main power supply and having volatile data registers; a Read/Write memory; a mains fail detector detecting the failure of the main power supply; a standby power supply for said Read/Write memory; and a data-save controller including a logic network defining a control unit effective upon the detection of a failure in the main power supply, to cause the transfer of the information in the volatile data registers of the processor to the Read/Write memory powered by the standby power supply and to monitor such transfer, and upon the detection of the restoration of the main power supply, to cause the retransfer of the information from the Read/Write memory to the volatile data registers of the processor and to monitor such retransfer.

In the preferred embodiment of the invention described below, the logic network of the data-save controller comprises: fixed memory devices decoding inputs from the data processor to the controller and defining the state sequence of the data-save controller; a plurality of flip-flops defining the existing state of the data-save controller; and a further flip-flop indicating whether transfer of said information from the volatile data registers to the Read/Write memory has been effected.

Preferably, the logic network of the data-save controller further includes a timer actuated by the restoration of the main power supply and effective to prevent the initializing of the data-save controller until after a predetermined time interval has elapsed from the actuation of said timer.

Further features and advantages of the invention will be apparent from the description below.

The invention is herein described, by way of example only, with reference to the accompanying drawings wherein:

Fig. 1 is a block diagram of one form of data processing system constructed in accordance with the invention;

Fig. 2 is a block diagram of the data-save control portion of the system of Fig. 1;

Fig. 3 is a logic diagram of the data-save controller in the system of Fig. 2;

Fig. 4 is a block diagram of one known form of processor which may be used in the system of Fig. 1;

Fig. 5 is a state diagram illustrating the sequential operation of the machine state control unit defined by the logic network in the data-save controller of Fig. 3; and

Figs. 6 and 6a illustrate one simplified program which may be used for performing an Interrupt routine in a Data-Save Operation.

A system employing the present invention is illustrated in generalized form in the block diagram of Fig. 1. It includes a processor 2 supplied from an AC main power supply 3. The processor 2 is of the programmable type, for example that described in detail in U.S. Patents 3,886,523, 3,930,236, and 4,005,391. The processor is adapted to communicate with a host of peripheral units, such as a card reader punch 4 and a printer 5 via a common bidirectional I/O (Input-Ouput) bus 6. In addition, the processor 2 communicates with an external Read/Write memory 7 via a memory interface 8. The memory address is supplied to the interface 8 via bus 10, and the Read/Write signal is fed via bus 12, the interface 8 in turn feeding to memory 7 an Enable signal via bus 13, a Read/Write address via bus 14, and the clocks via bus 16.

The specific construction and operation of the processor 2, and the manner of control of its peripheral units 3, 4, do not form a part of the present invention and therefore are not described herein in detail. A block diagram of a processor that may be used is illustrated in Fig. 4 and is described below to the extent necessary for an understanding of the present invention. If desired, further details of the construction and operation of such a processor may be had by reference to the above-cited patents.

The present invention concerns primarily the preservation of the volatile data in the registers of the processor 2 in the event of a power failure in the AC main power supply 3. For this purpose, a data-save control system, generally designated 20 in Fig. 1, is coupled to the processor 2 via its bidirectional bus 6. The data-save control system is supplied with isolated AC via line 21 from the processor 2. In addition, the data-save control system includes a DC supply 22 connected to it via line 24, and one or more batteries 26 connected to it via line 28, the latter line including a cut-out switch 30. Upon the detection of a failure in the AC main power supply 3 by a mains-fail detector within the data-save control system 20 as will be described more particularly below, the data-save control system is effective to monitor the transfer, via Memory Write bus 32, of the information in the volatile data registers of the processor 2 to the external Read/Write memory 7; and upon the detection of the restoration of the supply mains, the data-save control system 20 is effective to monitor the retransfer, via Memory Read bus 34, of the information from the Read/Write memory 7 to the volatile data registers of the processor. During the foregoing operation of the data-save control system 20, it supplies standby power, via bus 36, to the Read/Write memory 7 for a sufficient time to complete the transfer and retransfer operations.

Cut-out switch 30, in line 28 to the standby batteries 26, is actuated by a relay R1 in line 38, which relay is controlled by battery chargers included in the data-save control system 20. In addition, the data-save control system 20 controls, via line 39, a circuit breaker relay R2 having contacts 40 in the mains to the AC power supply 3, such as to inhibit the reconnection of processor 2 to the main power supply until a predetermined time interval has elapsed following the restoration of the AC main power supply. All the foregoing operations, and the manner the data-save controller performs them, are described more particularly below.

Fig. 2 is a block diagram of the data-save control system 20 of Fig. 1. The system comprises an instruction decoding unit 42, decoding the instructions received from the processor 2 via bus 6; a machine state control unit 44 including the logic network defining the state sequence of the controller; a first timer TC1; a mains-fail detector 46 detecting a failure (e.g., either blackout or brownout) in the AC main power supply 3 as sensed via the isolated AC input line 21; a latch 48 for latching the Mains-Fail signal received via line 47 from detector 46; a second timer TC2; and an output gating circuit 50 gating the output to I/O bus 6. The first timer TC1 is preferably a digital counter set to time-out 16 ms after actuation; it is used for timing in the state sequence. The second timer TC2 is preferably an R-C Schmidt trigger set to time-out about 0.5 to 2 seconds after actuation; it is used for restart timing. All

the foregoing components, which may be implemented on a single IC (Integrated Circuit) chip, constitute a Data-Save Controller, which Controller is described more particularly below with respect to Fig. 3.

The data-save control system 20 illustrated in Fig. 2 further comprises a battery charger 60 connected to the DC supply 22 via lead 24 and adapted to maintain the standby battery 26 (Fig. 1) fully charged by controlling cut-out switch 30 via its relay R1 in line 38; a battery under-voltage protection unit 62 guarding against the batteries becoming too deeply discharged; an inverter unit 64 for supplying the standby AC voltage to all the devices, including the external Read/Write memory 7 (Fig. 1) via line 36, during a Mains-Fail condition; an under-voltage protection unit 66 protecting inverter 64 against an under-voltage; an over-voltage protection unit 68 protecting the inverter against an over-voltage; and a drive circuit 70 which supplies driving current to circuit breaker R2 whose contacts 40 (Fig. 1) are in the AC supply mains.

The foregoing units, which can be implemented on one or more additional IC chips, may be of known construction, and therefore further details of their construction and operation are not deemed necessary.

Fig. 3 illustrates more particularly the data-save controller in the circuit of Fig. 2. The controller in Fig. 3 includes two fixed memory devices 72, 74, preferably PROM's (Programmable Read-Only Memory devices) which decode the inputs received from the processor 2 via I/O bus 6. The data-save controller further includes four J-K flip-flops 76, 78, 80, 82, three of which (76, 78, 80) define the existing state of the data-save controller, the remaining one (82) registering the success or otherwise of the Data-Save Operation. In addition, the controller includes the two previously mentioned timers TC1 and TC2 used for timing in the state sequence; the mains-fail latch 48 for latching the Mains-Fail signal received from the mains-fail detector 46 (Fig. 2) via line 47; the output gating circuit 50 for gating the output from the controller via I/O bus 6 to the processor 2; and a test flip-flop 84 which is used during a testing operation to reset the mains-fail latch 48.

As indicated earlier, the data-save control system of the present invention may be used with different types of processors, Fig. 4 illustrating, for purposes of example, one form of processor 2 with which the invention may be used, the illustrated processor being that more particularly described in the above-cited patents.

Briefly, the processor illustrated in Fig. 4 is one driven by micro-instructions made up of varying numbers of syllables, depending upon the function and literal values required. The processor employs two levels of subinstruction sets by which macro (or subject) instructions are implemented by strings of micro instructions all of which are implemented by control instructions. Each level of instruction sets may be stored in separate portions of memory, or even in separate memories, with the control instructions being stored in a Read-Only memory internal to the processor. The micro instructions are thus formed of varying numbers of syllables, with the different syllables being stored in a micro instructions memory and fetched in sequence under the control of a Machine State Control unit TMS to form the particular micro instruction. One syllable of each micro instruction is selected to indicate the particular combination of the function to be performed, the source and destination register to be employed, the particular busses which are to be used for data transfer, and the timing of micro instruction execution, i.e., the number of characters, digits, or bits to be operated upon during the micro instruction execution. When the particular micro instruction is formed of more than one syllable, the remaining syllables represent values or literals used as address parameters and also for logical operations.

As illustrated in Fig. 4, the processor 2 includes a function unit 120 to which data is supplied by A-bus 121 and B-bus 122, and from which data is received by F-bus 123. All data moves from the various registers through function unit 120. These respective busses are 8 bits wide, which is the basic width of all syllables and data segements employed in the system. A-bus 121 and B-bus 122 receive information segments from the respective registers, and also from memory, by way of U-buffer register 124, which is also employed to supply 8-bit addresses to control memory 137.

The machine instructions or S-instructions (which may be a higher level programme language, such as Cobol) are implemented by strings of micro instructions which are stored in an external memory. Preferably, a portion of external memory 7 (Fig. 1) is used for this purpose, the memory being divided into separate portions including a Read-Only portion for the permanent storage of micro instructions to provide "bootstrap" facilities, and Read/Write portions for storing the S-instructions, some micro instructions, and data during the the normal operation of the processor. In addition, the Read/Write portion of external memory 7 is used during the Data-Save operation for storing the

volatile data in internal registers of the processor upon the detection of a failure in the main power supply (3, Fig. 1), the information being retransferred back to the volatile data registers when the power supply has been restored, as will be described more particularly below.

The memory address registers MAR1 and MAR2 are identical 16-bit registers which operate in either the Transfer Mode or the Count Mode. In the Transfer Mode, each register is arranged as two 8-bit byte registers 125a, 125b and 126a, 126b, both capable of being loaded from function unit 120 by way of F-bus 123. When in the Count Mode, each of the memory address registers is employed to address memory via a 16-bit output bus 144 connected to the memory address bus 10 (Fig. 1).

The processor includes the following additional registers: BO-Register 127 and B1-Register 128, which are single character general purpose registers; B2-Register 129a and b3-Register 129b, which are single character general purpose registers that may be concentrated form a two-byte register; flag register 130, which is a single character register for storing general flags bytes; Y-Registers 131a—131d, and X-registers 133a—133d, which may, respectively, be concatenated together to form two 4-byte registers or one 8-byte (16 digit) registers (XY); working registers WRU, WR1; and additional registers JU, JL, KU, KL, and LU, LL.

The processor 2 further includes micro address registers (μMAR1—5) 135 capable of being loaded from, or unloaded to, function unit 120. They can be arranged to form a push-down, or last-in-first-out (LIFO), address stack for micro memory addressing and for storing program and interrupt routine addresses. This information is outputted via 16-bit micro memory address bus 145 and the memory address bus 10 (Fig. 1) to the memory interface unit 8.

In addition, processor 2 further includes U-buffer register 124, which is an 8-bit register used for addressing control memory 137 and for providing information about the next micro instruction to be executed. This information is used to generate overlap of the micro instruction fetch and execution phases. Upon the accessing of control memory 137, a control instruction is supplied to control buffer register 138, which holds the signals of a control instruction during the time required for its execution.

The input-ouput interface of the processor via I/O bus 6 comprises I/O address bus 143 connected to I/O address register 141, and I/O request bus 142. I/O address register 141 is an 8-bit register used to address a plurality of bi-directional I/O channels or control units, and is loaded from, or unloaded to, function unit 120.

Further details of the construction and operation of the processor may be had by reference to the above-cited patents. For purposes of the present invention, suffice it to point out that all the above-mentioned registers illustrated in Fig. 4 are internal to the processor; and that all, except those of the control memory 137 and the micro address registers 135, are adapted to contain volatile data which is to be saved in the event of a mains failure by the immediate and automatic transfer of the information from the volatile registers to the external Read-Write memory 7 supplied by the standby power supply via bus 36, the information being automatically transferred from the external memory 7 to the processor registers upon the restoration of the supply mains, the transfer and retransfer of such information during a Data-Save Operation being monitored by the data-save controller in the control system 20 of Figs. 1 and 2.

The state diagram of Fig. 5 illustrates the manner in which the data-save controller of Fig. 3 monitors the transfer of the information from the volatile registers in the processor- 2 to the external Read/Write memory 7 upon the detection of a mains failure, and the retransfer of the information back to the registers upon the restoration of the supply mains. As pointed out above, the state sequence of the data-save controller is defined by the logic network of the controller illustrated in Fig. 3.

State-0 is the initialized state of the controller, in which the data-save condition flip-flop 82 (Fig. 3) is reset, and the mains-fail latch 48 outputs a low level signal to fixed memory device 74. In State-0, the controller can be affected by:

1) the Mains-Fail signal from latch 48 going high, indicating that the mains-fail detector 46 (Fig. 2) has detected a failure (either black-out or brown-out) in the AC main power supply 3 (Fig. 1); or

2) Control Write signal being generated by the processor 2 with a specified data word.

In the event either of these conditions occurs, the controller moves to State-1, generating a Request for access to the memory, and in addition, actuating the delay counter TC1 which counter times-out at 16 ms. If the Mains-Fail signal goes high, it is latched by latch 48 and is thus synchronized to the system clocks in order to obviate timing errors.

The provision for moving the controller from State-0 to State-1 by a Control Write signal is a test facility to enable simulation of a "mains failure" from the processor.

State-1 is a Request State, in which the controller requests the processor for access to the memory. If, while the controller is in State-1 the Mains-Fail signal returns low indicating that the supply mains has been restored, the controller returns to State-0. As indicated above, the controller will also return to State-0 if the processor sends a control Write signal for testing purposes. In this case, the Control Write signal; sets the Mains Mail latch 48 via test flip-flop 84 (Fig. 3). The latter flip-flop is also actuatable by a Data In signal from the processor to reset the latch, this facility enabling the data-save operation to be inhibited during initial loading of the memories by causing the controller to toggle between States 1 and 0.

If, the controller, while in State-1 receives a Read Status signal from the processor, the controller moves to State-2. This is a transitional state, and starts timer TC1. If no recognisable change or signal is recorded during a time period of 16 ms, timer TC1 will run-out, and the controller will move to State-3.

If no Read Status is received from the processor while the controller is in State-1, timer TC1 will also run out after 16 ms, whereupon the controller will move directly from State-1 to State-3.

In State-3, the data-save controller generates another Request signal for access to the memory. If, when the responding Read Status signal from the processor is received by the Controller, the Mains-Fail signal has returned low (indicating that the main power supply has been restored), the controller will move to its initialized State-0. If, however, the Mains-Fail signal is still high, the Read Status signal will cause the controller to move to State-4. If no Read Status signal is received from the processor, the controller will lock-out in State-3.

Whenever there is a Request from the data-save controller, the processor reads the status, and all other interrupts are ignored. By reading the status, the processor can determine if the controller is in State-1 or State-3. A test flag is used for this purpose, the test flag being a dedicated bit in a register (e.g. flag register 130, Fig. 4) set or reset in order to differentiate between States 1 and 3. Thus, the test flag is set high after State-1, low after State-3, and is low in the other States 4, 5, 6, 7 and 0. If the data-save controller generates a Request signal before a data-save operation has been executed (this being recognized by the processor in the manner described below), the processor examines the state of the test flag. If the test flag is not set, the controller is in State-1; if the test flag is set, the controller is in State-3, and preparations must be made before shut-down.

While the controller is in State-4, the processor executes a stored data-save-operation program whereby it effects a transfer of the information from its volatile registers to a non-volatile portion of external memory 7 (Fig. 1). In State-4, the processor can also complete the current instruction and usually several more until a convenient point to stop is reached at which point it can then transfer the required registers to the memory for retention.

The data-save-controller remains in State-4 until it receives a Control Write Signal from the processor indicating that a successful Data-Save-Operation has been completed, i.e. that the information in its volatile registers has been transferred intact into the the non-volatile portion of external memory 7. If no such Control Write Signal is received by the controller, it will lock out in State-4. Upon receipt of Control Write signal indicating that a successful Data-Save Operation has been completed, the controller moves to State-5.

The data save controller remains in State-5 until the Mains-Fail signal from detector 46 and latch 48 returns low, indicating that power has been restored. All the while the Mains-Fail signal is high, the system remains in the Standby Mode, wherein the state flip-flops 76, 78, 80, 82 (Fig. 3) of the data-save controller, and the external memory 7 together with its refresh circuitry and clocks, are powered by the standby batteries 26 via line 36 (Fig. 1).

When the power is restored, the Mains-Fail signal goes low, but the data save controller will wait in State-5 and not move to State-6 until timer TC2 (Fig. 2) times out. As indicated above, this timer is an RC Schmidt trigger circuit. It may be preset to time-out after about 1 second and provided to ensure that start-up control is by the data-save controller. It prevents the possibility that the processor may examine a Request signal before the data-save controller has had a change to generate one in State 6, i.e., before the information from the memory has been restored into the appropriate registers by the Data-Save Operation. The timer is triggered by the Mains-Fail signal going high upon the restoration of the power to ensure the discharge of its timing capacitor, so that if power returns upon the controller entering State-5, the time-out will still have to run. It also ensures that on reaching State-5, the state flip-flops (76, 78, 80, 82, Fig. 3) can be locked in that state independently of clocks and input data, thereby reducing the number of devices required to be powered during the Standby mode. As soon as the time (e.g. 1 second) preset in timer TC2 has run out from the time the controller has moved to State-5 (and assuming the Mains-Fail signal has gone low, indicating power

has been restored), the controller moves to State-6.

When the data-save-controller is used with the processor illustrated in Fig. 4, the first address register μMAR1 of its micro memory address store 135 is reset, and the processor executes a programme stored in a non-volatile memory portion of the external memory 7, whenever the power is switched on, or a Request is generated by a controller indicating a requirement for updated information or similar action by the processor. The programme executed by the processor first instructs the processor to store the address of the location, held in the memory address register (μMAR1) before interruption, in a subsidiary register (μMAR3), so that the processor can return to that address after servicing the Interrupt on receipt of an Enable Return signal. If the system has only been switched on, there being no volatile data to be saved, no Enable Return signal will be executed, and instead, the address of the first instruction required in the volatile store will be entered into the first address register (μMAR1) of the micro address store 135.

After temporarily storing the original memory address, the programme stored in the non-volatile memory determines if there is a Request from the data-save controller. If there is no request, indicating that no data has been saved, the processor generates a Read Status signal moving the data-save controller from State-6 to State-7. A control Write signal is then sent from the processor, which moves the controller from State7 to the initialized State-0.

If the mains should fail while the controller is in State-7, the controller will immediately return to State-5.

As indicated earlier, the processor can identify either of the Request States 1 or 3 by examining the Test Flag (e.g. in flag register 130, Fig. 4), the Test Flag being reset in State-1 and set in State-3.

Both of the above States 1 or 3 can occur only when the Mains-Fail signal is high, indicating a failure in the supply mains. If Mains-Fail signal is low, however, indicating that the power has been restored, and a Request is generated by the controller, this means that the controller is in State-6. In this state, the data-save condition flip-flop 82 (Fig. 3) is set, indicating that the volatile register information of the processor is stored in the non-volatile portion of its external memory 7. The processor thereupon retransfers the volatile information from external memory 7 back to the volatile registers within the processor, and generates a Control Write signal, resetting the data-save-condition flip-flip 82, and moving the controller from State-7 back to its initialized State-0.

Now, with the data-save controller initialized, other Requests can be serviced.

Fig. 6 and 6a illustrate a simple programme that may be used with the described data-save controller for performing an Interrupt routine upon receiving a Request from the data-save controller or from any other peripheral unit.

Breifly, if the power is switched on, or a Request for access to the memory (through the processor) is received, the memory address register (μMAR1, Fig. 4) is reset, and the machine executes a programmed stored in the non-volatile ROM memory (137, Fig. 4). The latter programme first instructs the machine to store, in a subsidiary register (μMAR3), the address of the location held in the memory address register (μMAR1) before the Interrupt, so that the machine can return to that address after servicing the Interrupt on receipt of the Enable Return. If the machine has only been switched on, no data having been saved, no Enable Return will be executed and, instead, the address of the first instruction required in the volatile store will be entered into the address register (μMAR1).

Thus, with reference to Fig. 6, upon receiving the Interrupt (block 200) and after temporarily storing the original memory address (block 202), the processor determines if there is a Request from the data save controller (block 204). If there is no Request, indicating that no data had been saved, the processor generates a Read Status command (block 206) moving the Data-Save State Machine from State-6 to State-7. (As described earlier, the transition from State-5 to State-6 is automatic if the mains has returned (MFail low) and the time constant TC2 has elapsed). A Control Write command (block 208) is then issued which moves the controller from State-7 to the initialized State-0. Now, with the data save controller initialized, other Requsts can be serviced, or the machine registers can be initialized, i.e. set to a known state enabling software execution (block 210).

If there is a Request from the data-save controller (block 204), all other Interrupts are ignored (block 212) and a Read Status command is issued (block 214) to enable the processor to determine whether the controller is in State-1 or State-3. Both of these states are characterized by the outputs 0101; if the output is not 0101 (determined in block 216) but rather is 0100 (determined in block 218), this indicates that the controller is in State-1 or State-3 but the mains have returned (MFail is low). In such a case, the controller will have been reset to its initialized State-0 as described above,

and the software now returns the operation to its original programme, by resetting the Test Flag (block 220), enabling other Interrupts (block 222), An Enable Return is then executed (block 224), returning the original programme address from μMAR3 to μMAR1.

If the output received as a result of the Read Status command (block 214) is neither 0101 (block 216) nor 0100 (block 218), the only other Request State is State-6 indicating the register contents are store in the memory by the setting of the Data Save Condition Bit (DSCB), i.e. flip-flop 82, Fig. 3. The processor then reloads the registers (block 226) and generates a Control Write (block 228) resetting the DSCB flip-flop 82 and moving the controller from State-7 to the initialized State-0. The Test Flag is reset (block 220); the Interrupts are enabled (block 222); and an Enable Return (block 224) is executed returning the machine to the control of software.

On the other hand, if the output received as a result of the Read Status command (block 214) is 0101, this indicates that the controller is either in State-1 or State-3. As described above, the Test Flag, which is a dedicated bit in a register (e.g. flag register 130, Fig. 4), is used for distinguishing between State-1 and State-3. Thus, the condition of the Test Flag is examined (block 230), and if it is found that it is not set, the controller is in State-1; the Test Flag is then set (block 231). If the Test Flag is found to be set, the controller is in State-3, and preparations must be made before shut-down.

The Test Flag is reset (block 232); the relevant registers are stored (block 234); and a Control Write command is generated (block 236) bringing the controller to State-5. A dynamic loop is then set up awaiting shut-down.

Fig. 6a illustrates the software the software loop which may be used in this example. It is a simple counting loop using the XY-register (namely the X-register and the Y-register concatenated to make a 16-bit register), and the B0 and B1 registers. Counting into the XY-register acts as a delay between increments to the B0-register. The B1-register is used because of association and easy access to a display peripheral unit.

When interruption to the counting occurs and it is the result of a mains failure, the contents of the XY, B1 and B2 registers are stored in the memory. In this case, it is not necessary to use the registers to hold the count. After each increment, the data could be stored in a dedicated location, thus obviating the need to transfer the data to the store in the event of a shut-down. It should be noted that transfer of information from the registers to store is achieved by the use of programme codes opening data paths and writing that data into locations, the addresses of which are again set up by software instructions. Recall of the data is achieved by the same method.

The operation of the data-save controller requires that the computer be fully powered by capacitive means for at least 20 ms after the failure, partial or total, of the mains supply. This allows 4 ms to complete computations and information transfer to the memory.

It is possible, theoretically, to save the data in every register in the processor if required, subject to the limitation that in the described embodiment there are only 4 ms available to do this. However, the normal practise would be to interrupt the operation at a point where the minimum number of registers will be required to be transferred to the store (external memory 7, Fig. 1) for retention.

While the invention has been described with respect to one preferred embodiment, it will be appreciated that many variations, modifications and other implementations of the invention may be made.

WHAT WE CLAIM IS:—

1. A data processing system comprising:

a processor supplied with power from a main power supply and having volatile data registers; a Read/Write memory;

a mains fail detector detecting the failure of the main power supply; a standby power supply for said Read/Write memory: and a data-save controller including a logic network defining a control unit effective

upon the detection of a failure in the main power supply, to cause the transfer of the information in the volatile data registers of the processor to the Read/Write memory powered by the standby power supply and to monitor such transfer, and

upon the detection of the restoration of the main power supply, to cause the retransfer of the information from the Read/Write memory to the volatile data registers of the processor and to monitor such retransfer.

2. A system according to Claim 1, wherein said logic network of the control unit comprises:

fixed memory devices decoding inputs from the data processor to the controller and defining the state sequence of the data-save controller;

a plurality of flip-flops defining the existing state of the data-save controller;

and a further flip-flop indicating whether transfer of said information from the volatile data registers to the Read/Write memory has been effected.

3. A system according to Claim 2, wherein

said logic network of the control unit includes four flip-flops defining a sequence of eight states.

4. A system according to either of Claims 2 or 3, wherein said logic network of the data-save controller further incudes a timer actuated by the restoration of the main power supply and effective to prevent the initializing of the data-save controller until after a predetermined time interval has elapsed from the actuation of said timer.

5. A system according to Claim 4, wherein said timer includes an R-C network set to time-out from 0.5 to 2 seconds after its actuation.

6. A system according to any one of Claims 2—5, wherein said logic network of the data-save controller further includes a digital-counter timer for monitoring the transfer of the information in the volatile data register of the data processor to the Read-Write memory.

7. A system according to any one of Claims 1—6, wherein said Read/Write memory is external to the processor.

8. A system according to any one of Claims 1—7, wherein said data-save controller further includes a latch for latching the output signal from the mains fail detector.

9. A system according to any one of Claims 1—8, wherein said standby power supply includes a standby battery and an inverter powered by the battery and supplying the standby power to the Read/Write memory.

10. A system according to Claim 9, wherein said standby power supply further includes a battery charger for charging said standby battery, a battery under-voltage detector controlling the charging of said battery, and an inverter supply protection circuit including over-voltage and under-voltage detectors controlling the inverter powered by the battery.

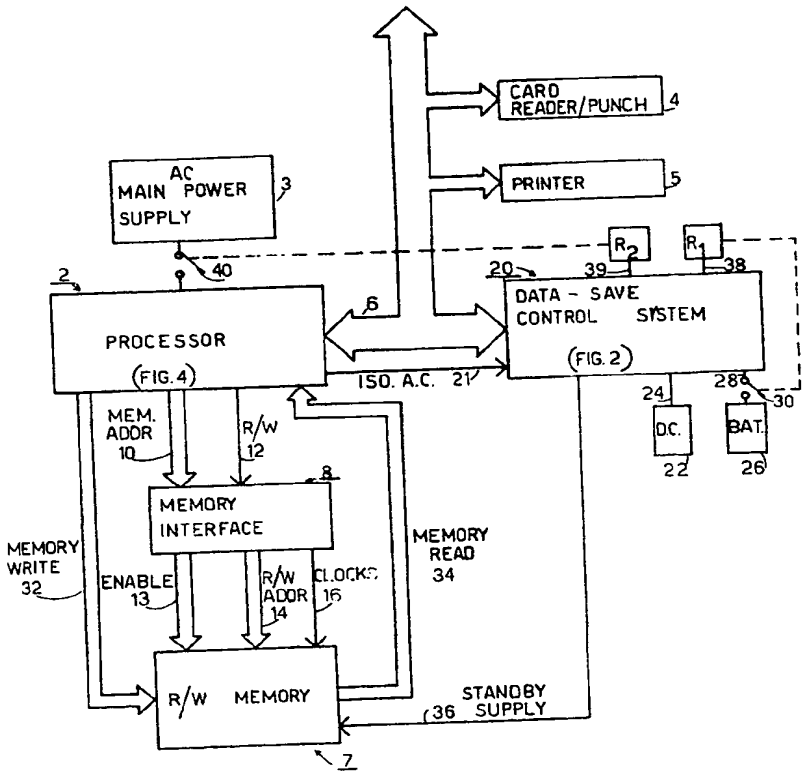
11. A data processing system including a data-save controller substantially as described with reference to and as illustrated in the accompanying drawings.

For the Applicants,
G. F. REDFERN & COMPANY,
Marlborough Lodge,
14 Farncombe Road,
Worthing,
West Sussex BN11 2BT.

Printed for Her Majesty's Stationery Office, by the Courier Press, Leamington Spa, 1979
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

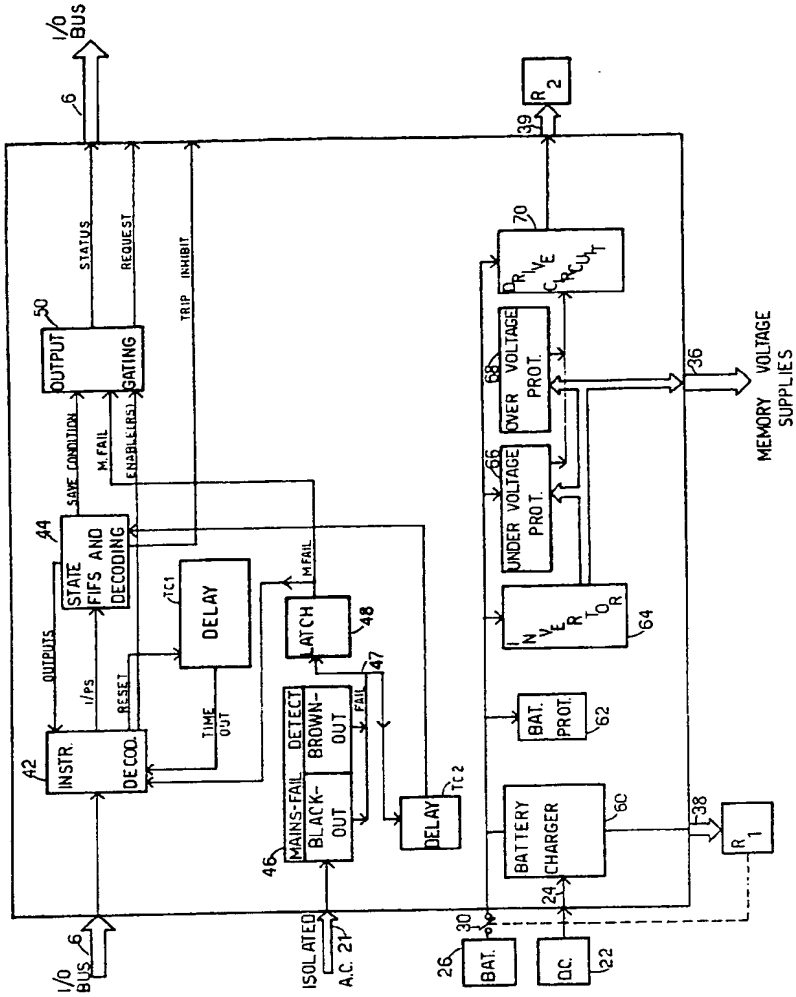
1545169 COMPLETE SPECIFICATION
7 SHEETS *This drawing is a reproduction of the Original on a reduced scale*
Sheet 1

FIG.1



1545169 COMPLETE SPECIFICATION
7 SHEETS *This drawing is a reproduction of the Original on a reduced scale*
Sheet 2

FIG. 2



1545169 COMPLETE SPECIFICATION
7 SHEETS This drawing is a reproduction of the Original on a reduced scale
Sheet 3

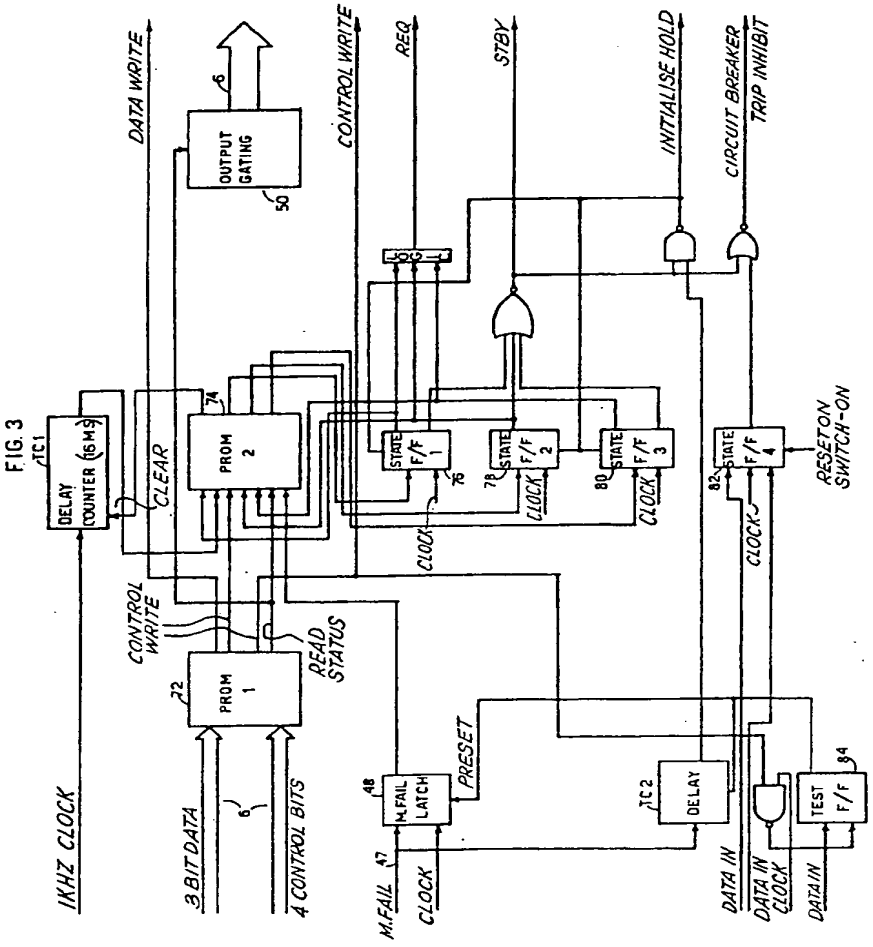
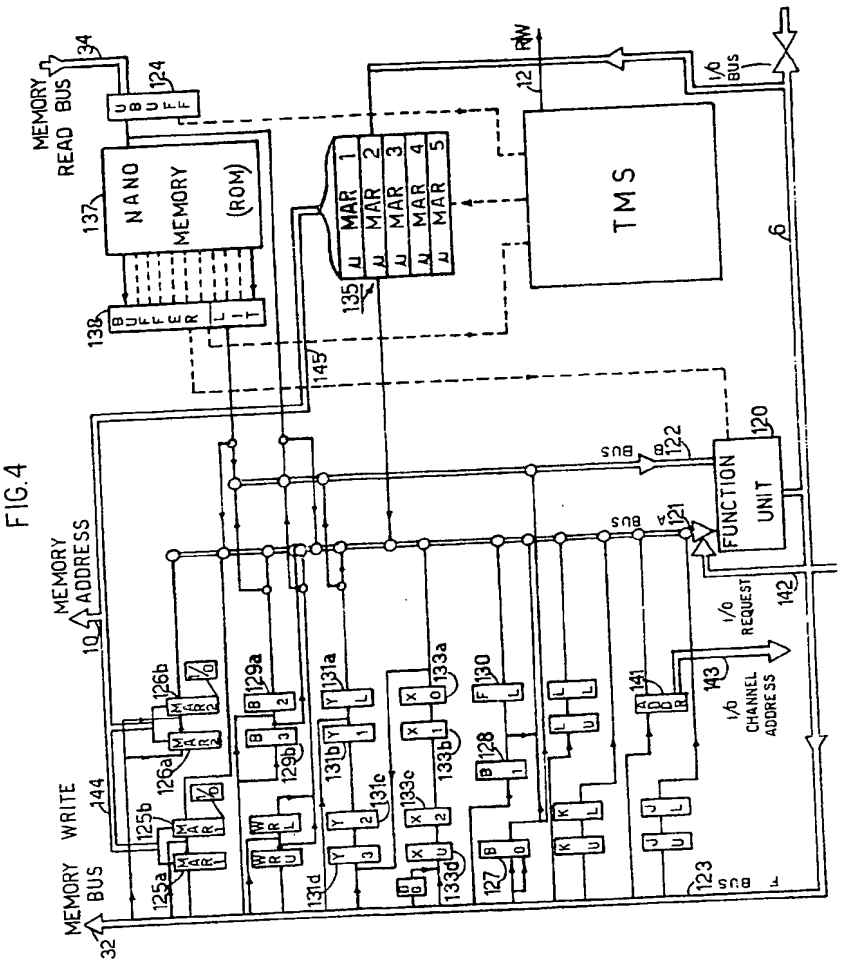


FIG. 3

1545169 COMPLETE SPECIFICATION
7 SHEETS This drawing is a reproduction of the Original on a reduced scale
Sheet 4

FIG.4



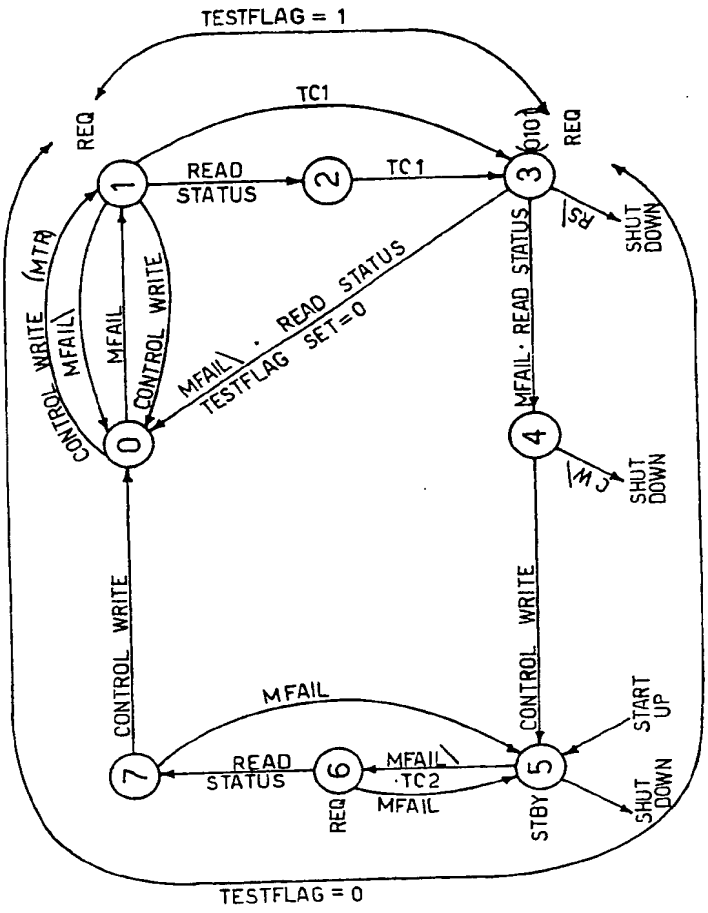

TESTFLAG = 1
TC1
REQ
READ STATUS
TC1
(010)
REQ
SHUT DOWN
RS
CONTROL WRITE (MTA)
MFAIL
MFAIL
CONTROL WRITE
MFAIL · READ STATUS
TESTFLAG SET=0
MFAIL · READ STATUS
SHUT DOWN
MJ
CONTROL WRITE
CONTROL WRITE
MFAIL
START UP
READ STATUS
MFAIL ·TC2
MFAIL
REQ
STBY
SHUT DOWN
TESTFLAG = 0

FIG. 5

FIG. 6

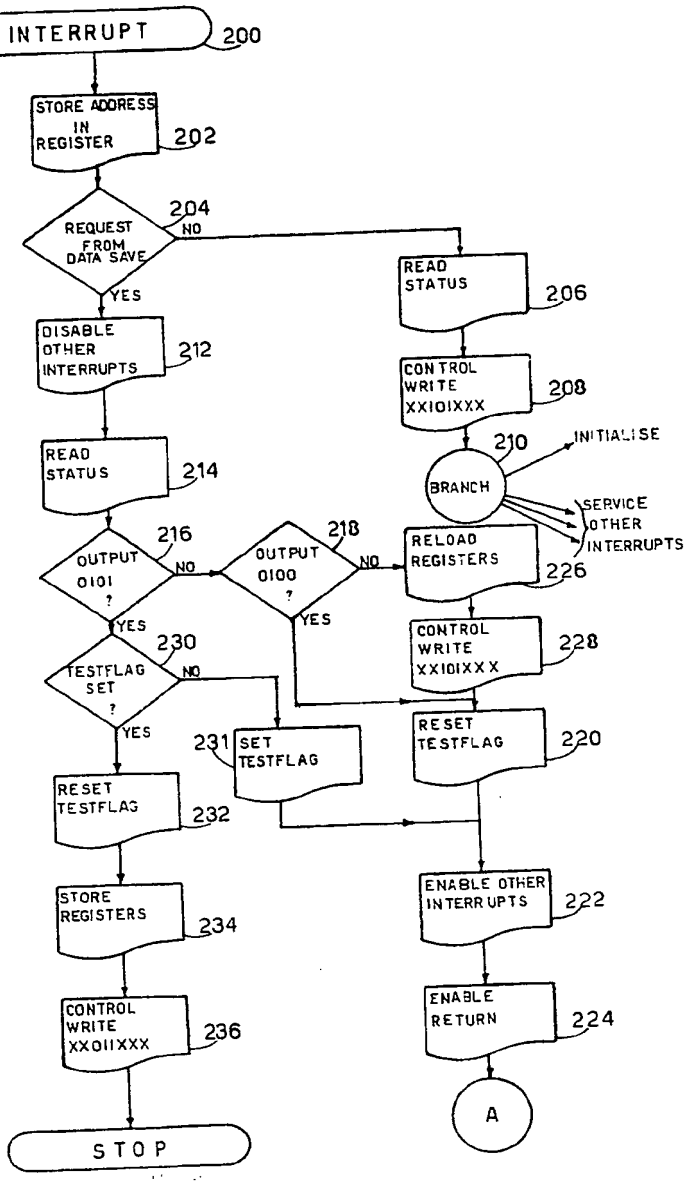
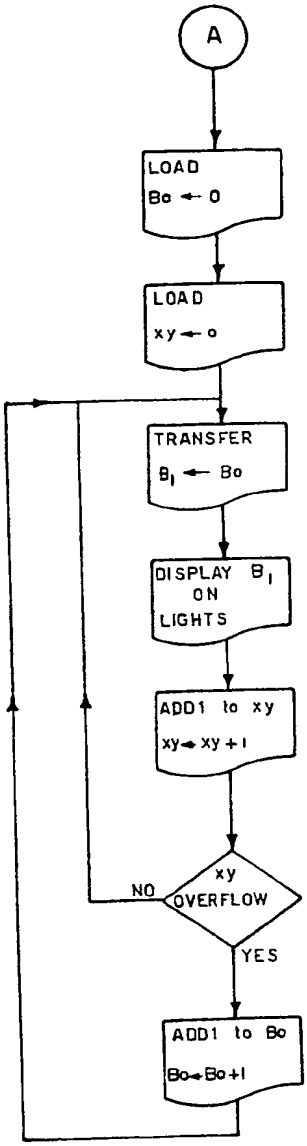

INTERRUPT
200
STORE ADDRESS IN REGISTER
202
REQUEST FROM DATA SAVE
204
NO
YES
READ STATUS
206
CONTROL WRITE XX10IXXX
208
INITIALISE
210
BRANCH
SERVICE OTHER INTERRUPTS
DISABLE OTHER INTERRUPTS
212
READ STATUS
214
OUTPUT 0101 ?
216
NO
YES
OUTPUT 0100 ?
218
NO
YES
RELOAD REGISTERS
226
CONTROL WRITE XX101XXX
228
TESTFLAG SET ?
230
NO
YES
SET TESTFLAG
231
RESET TESTFLAG
220
RESET TESTFLAG
232
ENABLE OTHER INTERRUPTS
222
STORE REGISTERS
234
ENABLE RETURN
224
CONTROL WRITE XX011XXX
236
A
STOP

FIG. 6a


A
LOAD
Bo ← 0
LOAD
xy ← o
TRANSFER
B1 ← Bo
DISPLAY B1
ON
LIGHTS
ADD1 to xy
xy ← xy + 1
xy
OVERFLOW
NO
YES
ADD1 to Bo
Bo ← Bo+1